

REMARKS

The present reply is in response to the final Office Action of September 8, 2006. Claims 28-45 were allowed in the Office Action, and claim 4 was objected to as depending from a rejected base claim. Claims 1-3, 5-8, 10-18, 20-27, 46, 48-55 and 57-63 were rejected. Claims 1-8, 10-18, 20-46, 48-55 and 57-63 are again presented for the Examiner's consideration in view of the following remarks.

Reexamination and reconsideration of the above-identified application, pursuant to and consistent with 37 C.F.R. § 1.116, and in light of the remarks that follow, are respectfully requested. Because the present claims are believed to be in condition for allowance over the cited art, good cause exists for the entry of this reply in accordance with 37 C.F.R. § 1.116.

As an initial matter, the applicants would like to thank the Examiner for conducting the telephone interview on October 10, 2006 with the undersigned. The telephone interview was most helpful in clarifying the issues presently in this case.

In the Office Action, claims 8-17 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Per the telephone interview, the Examiner has agreed to withdraw this rejection. Therefore, for purposes of this reply, the § 101 rejection will be treated as withdrawn.

Claims 1-3, 5-8, 10-18, 20-27, 46, 48-55 and 57-63 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,421,736 ("*Breslau*"). Applicants respectfully traverse the rejection.

As discussed in the amendment filed by applicants on June 1, 2006, *Breslau* is directed to an object-oriented computer system that manages instances of objects, in particular

migration of objects "between a merged status having a single instance and a split status having multiple instances." (Col. 1, ll. 29-31.)

Numerous deficiencies of *Breslau* were explained in detail in the June 1, 2006 amendment filed in response to the previous Office Action of March 7, 2006. For the sake of brevity, those detailed explanations will not be repeated herein, although they are incorporated by reference. Notwithstanding those deficiencies of *Breslau*, which applicants submit render the rejection of the claims improper, applicants will now address particular features of software cells of the claimed invention, as presented in independent claims 1, 8, 18, 46 and 55.

In particular, independent claim 1 recites "each of said software cells comprising a program compatible with said instruction set architecture, data associated with said program, information for routing said software cell over said network, and an identification number uniquely identifying said software cell among all of said software cells being transmitted over said network."

Independent claim 8 recites "said software cell comprising: a program for processing by one or more of said processors; data associated with said program; information for routing said software cell over said network; and a global identification uniquely identifying said software cell among all software cells being transmitted over said network."

Independent claim 18 recites "each of the software cells comprising: a program for processing by one or more of said processors; data associated with said program; information for routing said software cell over said network; and a global

identification uniquely identifying said software cell among all software cells being transmitted over said network."

Independent claim 46 recites "each of the software cells comprising: a program for processing by said at least one processor; data associated with said program; information for routing said software cell over the computer network; and a global identification uniquely identifying said software cell among all software cells being transmitted over the network."

And independent claim 55 recites "each of the software cells comprising: a program for processing by one of said processing unit and said at least one sub-processing unit; data associated with said program; information for routing said software cell over the computer network; and a global identification uniquely identifying said software cell among all software cells being transmitted over the network."

In each of these independent claims, it is the software cell which comprises, among other features, information for routing the software cell over the network. The Office Action asserts that *Breslau* teaches such a feature, and equates the "objects" in *Breslau* with the claimed software cells. Specifically:

Routing of objects was clearly possible in the disclosure of *Breslau*, such routing being based on the identity of the object (see column 8, lines 4-54). As described above, each object comprises a unique ID, and as such, routing based on such an identity clearly teaches that the software cell included information for routing the cell over the network as broadly claimed.

Office Action, pg. 10, numbered section 36.

Applicants strongly disagree with this assessment of *Breslau*. According to the cited portion of *Breslau*, the reference discloses "ORB table management." An "ORB" is an

Object Request Broker, "which is responsible for, e.g., creation, deletion and tracking of objects." (Col. 4, ll. 13-14.) As stated in Breslau:

ORB table management, according to the techniques of the present invention, is discussed below. Conventionally, **an ORB Routing Table tracks objects registered within the object-oriented computer system.** Included in this table is an identifier for each object, along with pertinent information associated therewith. For example, the name of the particular execution environment the object is loaded within is stored in the table along with each object's identifier. According to the techniques of the present invention, modification to the ORB routing table is performed to support objects having split and merged status.

Shown in FIG. 6 is an example of an ORB Routing Table corresponding to the computer system configuration depicted in FIG. 2. Each entry within the table includes an object identifier ("OBJECT ID") and the execution environment that the corresponding object instance is loaded within ("LOCATION"). By way of example, each object instance shown in FIG. 2 is represented in the ORB Routing Table of FIG. 6. In particular, object A 53 is within host 37, object C145 is within workstation A 31, object C247 is within workstation B 33, object C349 is within workstation C 35 and object B 51 is within host 37. A "splittable" field is included for each object identifier and indicates whether the object can be split using the techniques disclosed herein. The "splittable" indicator is set by the object as part of the registration with the ORB. For example, it may be passed as a parameter to the ORB registration function. Also, a Split Instance Pointer ("SIP") points to a secondary routing table for split instances. If the object is not split, a null is stored in the SIP.

The ORB Routing Table of FIG. 7 corresponds to the system configuration of FIG. 3, wherein "object B" has been split into multiple instances thereof, namely objects B161, B263 and B365. The techniques of the present invention track these multiple instances using a "Split Routing Table" for each split object. As

shown in FIG. 7, the object identifier "Object_B" is associated with a SIP corresponding to a Split Routing Table for Object B. Each entry within this Split Routing Table contains the name of each "object B" instance, along with the name of the execution environment that the instance is within. Specifically, object B161 is within workstation A 31, object B263 is within workstation B 33 and object B365 is within workstation C 35. Thus, using these tables in connection with conventional ORB functionality, the identification, tracking and instantiation of the "split" instances of an object, e.g., "object B," are facilitated. If the merged instance of object B were maintained, the identifier "object B" would be maintained in the ORB routing table location field denoting a "partially split" state of object B.

(Col. 8, ll. 4-54, emphasis added.)

Even assuming, *arguendo*, that *Breslau's* objects (or even instances of objects) could be construed to be equivalent to "software cells," which applicants do not believe is the case, *Breslau* simply does not teach or suggest that the ORB routing table (or even that any of the information therein) is part of the objects (or instances of objects). While the ORB routing table tracks different objects and whether they are "splittable" or not, the table is discussed as being distinct from the objects themselves. See, e.g., col. 2, ll.38-48. It does not even appear possible that the table could be part of the object or instances of the object, because the table contains information for tracking the different objects in the system. See FIGS. 6-7.

During the telephone interview, the Examiner also identified another portion of *Breslau* as being pertinent to the claimed routing information, namely column 6 at lines 4-20, which states:

The process begins with the generation and/or receipt of a "split" request (STEP 61). This may be generated by another program or object executing

within the object-oriented computer system. As one example, the split request may be generated by a performance monitoring program that has determined an overload condition for a "merged" object within an execution environment. For example, with respect to the example object-oriented computer system of FIGS. 2-3, a performance monitor may detect that the single instance, object B 51 is overloaded on host 37, and accordingly generates a "split" request therefor, to predetermined execution environments determined by the heuristics of the performance monitor. A parameter list is passed from the performance monitor to the current process that enumerates the locations where the split instances are to be placed. The parameter list also indicates whether the original "merged" instance is kept intact.

Col. 6, 11.4-20.

As explained in the above-quoted portion of *Breslau*, the "parameter list" with "the locations where the split instances are to be placed" is passed to the process by the performance monitor. However, the process is not the object itself, nor is it one of the instances of the object. Rather, the process, as explained in the patent, is a procedure "for migrating an object from a merged status (i.e., having a single instance) to a split status (i.e., having multiple instances)." (Col. 5, 11.63-65.) The process has numerous steps, as illustrated in the flow diagram of FIG. 4. The process is simply inapplicable to the claimed invention, which **requires the software cell to comprise information for routing** the software cell over the network.

Thus, for at least these reasons, applicants submit that *Breslau* does not teach all of the elements of independent claims 1, 8, 18, 46 and 55. Applicants respectfully request, therefore, that the rejection of independent claims 1, 8, 18, 46 and 55 be withdrawn.

Claims 2-3, 5-6, 10-17, 20-27, 48-54 and 57-63 depend from independent claims 1, 8, 18, 46 and 55, respectively, and contain all the limitations thereof. Applicants respectfully request, therefore, that the rejection of the subject dependent claims also be withdrawn.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to withdraw the outstanding rejection of the claims and to pass this application to issue.

If, however, for any reason the Examiner does not believe that such action can be taken at this time, it is respectfully requested that he telephone applicants' attorney at (908) 654-5000 in order to overcome any additional objections which he might have. If there are any additional charges in connection with this requested amendment, the Examiner is authorized to charge Deposit Account No. 12-1095 therefor.

Dated: November 7, 2006

Respectfully submitted,

By 

Andrew T. Zidel

Registration No.: 45,256

LERNER, DAVID, LITTENBERG,

KRUMHOLZ & MENTLIK, LLP

600 South Avenue West

Westfield, New Jersey 07090

(908) 654-5000

Attorney for Applicant

699845_1.DOC